

#3  
10-18-01

PATENT

PEARNE & GORDON LLP  
526 Superior Avenue East  
Suite 1200  
Cleveland, Ohio 44114-1484  
(216) 579-1700



Attorney Docket No.: 33587

Box PATENT APPLICATION  
Assistant Commissioner for Patents  
Washington, D.C. 20231

Sir/Madam:

Transmitted herewith for filing by other than a small entity is the patent application of:

Inventor: Atsushi Ishiwata

For: "INTERMEDIATE OBJECT LINKING METHOD AND UNIT"

6 sheets of formal drawings are included.

An assignment of the invention to Matsushita Electric Industrial Co., Ltd. is included along with a Recordation Form Cover Sheet. Please record and return the assignment to the undersigned.

Priority is claimed under 35 U.S.C. § 119 on the basis of the following foreign application: **Japanese Patent Application No. 2000-238729 filed August 7, 2000.** A certified copy of this application is enclosed.

An Information Disclosure Statement is enclosed.

"Express Mail" mailing label number	EL653124595US
Date of Deposit	7-26-2001
I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. § 1.10 on the date indicated above and is addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.	
<u>Amanda Wittine</u> Printed Name of Person Mailing Paper or Fee	
<u>Amanda Wittine</u> Signature of Person Mailing Paper or Fee	

**CLAIMS AS FILED**

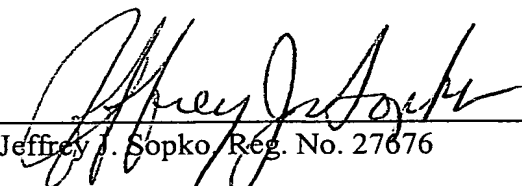
<u>For</u>	<u>Number</u>	<u>Rate</u>	<u>Fees</u>
Total claims in excess of 20:	0	18	\$0.00
Independent claims in excess of 3:	2	80	\$160.00
Multiple dependent claims, if any, add surcharge of \$270.00			\$0.00
Non-English Specification, add surcharge of \$130.00			\$0.00
		Basic Fee	<u>\$710.00</u>
Assignment Recordal Fee of \$40.00			\$40.00
		<b>Total Fee</b>	<b>\$910.00</b>

A check in the amount of the Total Fee calculated above is enclosed.

The Commissioner is hereby authorized to charge any fees under 37 C.F.R. §§ 1.16 and 1.17 which may be required during the entire pendency of this application, or to credit any overpayment, to Deposit Account No. 16-0820, Order No. 33587.

Respectfully,

**PEARNE & GORDON LLP**

  
\_\_\_\_\_  
Jeffrey J. Sopko Reg. No. 27676

Date: July 26, 2001

日 本 国 特 許 庁

PATENT OFFICE  
JAPANESE GOVERNMENT

J1040 U.S. PRO  
09/916168



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

2000年 8月 7日

出 願 番 号

Application Number:

特願2000-238729

出 願 人

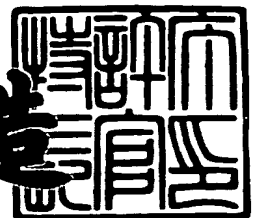
Applicant (s):

松下電器産業株式会社

2000年12月15日

特許庁長官  
Commissioner,  
Patent Office

及 川 耕 造



出証番号 出証特2000-3104167

【書類名】 特許願

【整理番号】 5038020001

【提出日】 平成12年 8月 7日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 9/06

【発明者】

    【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

    【氏名】 石渡 豊史

【特許出願人】

    【識別番号】 000005821

    【氏名又は名称】 松下電器産業株式会社

【代理人】

    【識別番号】 100105647

    【弁理士】

    【氏名又は名称】 小栗 昌平

    【電話番号】 03-5561-3990

【選任した代理人】

    【識別番号】 100105474

    【弁理士】

    【氏名又は名称】 本多 弘徳

    【電話番号】 03-5561-3990

【選任した代理人】

    【識別番号】 100108589

    【弁理士】

    【氏名又は名称】 市川 利光

    【電話番号】 03-5561-3990

【選任した代理人】

    【識別番号】 100115107

【弁理士】

【氏名又は名称】 高松 猛

【電話番号】 03-5561-3990

【選任した代理人】

【識別番号】 100090343

【弁理士】

【氏名又は名称】 栗宇 百合子

【電話番号】 03-5561-3990

【手数料の表示】

【予納台帳番号】 092740

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0002926

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 中間オブジェクト連結方法、及び、中間オブジェクト連結装置、及び、リンカ装置、及び、コンパイラドライバ装置、並びに、中間オブジェクトを連結するプログラムを記録した記憶媒体

【特許請求の範囲】

【請求項 1】 複数の中間オブジェクトを連結し、実行可能オブジェクトを作成する中間オブジェクト連結方法において、

前記複数の中間オブジェクトの連結順序を決定する中間オブジェクト連結順序作成ステップと、

前記中間オブジェクト連結順序作成ステップにおいて決定された連結順序に基づき前記複数の中間オブジェクトを連結処理して実行可能オブジェクトを得るリンク処理ステップと、

前記リンク処理ステップによって得られた前記実行可能オブジェクトのプログラムサイズを前記連結順序が変更される毎に直前の前記連結順序の前記プログラムサイズと比較する比較ステップと、

前記比較ステップによる比較の結果、前記プログラムサイズの小さい方の前記プログラムサイズとその時の連結順序とを更新して記憶する記憶ステップと、

前記中間オブジェクト連結順序作成ステップにより前記連結順序を変更し、前記リンク処理ステップと前記比較ステップと前記記憶ステップとを繰り返し実行する繰り返しステップとを備えたことを特徴とする中間オブジェクト連結方法。

【請求項 2】 前記中間オブジェクト連結順序作成ステップは、順列アルゴリズムによって前記連結順序を決定することを特徴とする請求項 1 記載の中間オブジェクト連結方法。

【請求項 3】 前記中間オブジェクト連結順序作成ステップは、遺伝的アルゴリズムによって前記連結順序を決定することを特徴とする請求項 1 記載の中間オブジェクト連結方法。

【請求項 4】 複数の中間オブジェクトを連結し、実行可能オブジェクトを作成する中間オブジェクト連結装置において、

前記複数の中間オブジェクトの連結順序を決定する中間オブジェクト連結順序

作成手段と、

前記中間オブジェクト連結順序作成手段において決定された連結順序に基づき前記複数の中間オブジェクトを連結処理して実行可能オブジェクトを作成するリンク起動手段と、

前記リンク起動手段によって作成された前記実行可能オブジェクトのプログラムサイズを前記連結順序が変更される毎に直前の前記連結順序の前記プログラムサイズと比較する比較手段と、

前記比較手段による比較の結果、前記プログラムサイズの小さい方の前記プログラムサイズとその時の連結順序とを更新して記憶する記憶手段と、

前記中間オブジェクト連結順序作成手段により前記連結順序を変更し、前記リンク起動手段と前記比較手段と前記記憶手段とを繰り返し実行する繰り返し手段とを備えたことを特徴とする中間オブジェクト連結装置。

【請求項 5】 前記中間オブジェクト連結順序作成手段は、順列アルゴリズムによって前記連結順序を決定することを特徴とする請求項 4 記載の中間オブジェクト連結装置。

【請求項 6】 前記中間オブジェクト連結順序作成手段は、遺伝的アルゴリズムによって前記連結順序を決定することを特徴とする請求項 4 記載の中間オブジェクト連結装置。

【請求項 7】 複数の中間オブジェクトを連結し、実行可能オブジェクトを作成するリンク装置において、

前記複数の中間オブジェクトの連結順序を決定する中間オブジェクト連結順序作成手段と、

前記中間オブジェクト連結順序作成手段において決定された連結順序に基づき前記複数の中間オブジェクトを連結処理して実行可能オブジェクトを作成するリンク処理手段と、

前記リンク処理手段によって作成された前記実行可能オブジェクトのプログラムサイズを前記連結順序が変更される毎に直前の前記連結順序の前記プログラムサイズと比較する比較手段と、

前記比較手段による比較の結果、前記プログラムサイズの小さい方の前記プロ

グラムサイズとその時の連結順序とを更新して記憶する記憶手段と、

前記中間オブジェクト連結順序作成手段により前記連結順序を変更し、前記リンク処理手段と前記比較手段と前記記憶手段とを繰り返し実行する繰り返し手段とを備えたことを特徴とするリンク装置。

【請求項 8】 前記中間オブジェクト連結順序作成手段は、順列アルゴリズムによって前記連結順序を決定することを特徴とする請求項 7 記載のリンク装置。

【請求項 9】 前記中間オブジェクト連結順序作成手段は、遺伝的アルゴリズムによって前記連結順序を決定することを特徴とする請求項 7 記載のリンク装置。

【請求項 10】 コンパイラ、アセンブラ、リンク等を起動することによって、ソースプログラムを翻訳し実行可能オブジェクトの作成を行うコンパイラドライバ装置において、

前記複数の中間オブジェクトの連結順序を決定する中間オブジェクト連結順序作成手段と、

前記中間オブジェクト連結順序作成手段において決定された連結順序に基づき前記複数の中間オブジェクトを連結処理して実行可能オブジェクトを作成するリンク起動手段と、

前記リンク起動手段によって作成された前記実行可能オブジェクトのプログラムサイズを前記連結順序が変更される毎に直前の前記連結順序の前記プログラムサイズと比較する比較手段と、

前記比較手段による比較の結果、前記プログラムサイズの小さい方の前記プログラムサイズとその時の連結順序とを更新して記憶する記憶手段と、

前記中間オブジェクト連結順序作成手段により前記連結順序を変更し、前記リンク起動手段と前記比較手段と前記記憶手段とを繰り返し実行する繰り返し手段とを備えたことを特徴とするコンパイラドライバ装置。

【請求項 11】 前記中間オブジェクト連結順序作成手段は、順列アルゴリズムによって前記連結順序を決定することを特徴とする請求項 10 記載のコンパイラドライバ装置。



【請求項 1 2】 前記中間オブジェクト連結順序作成手段は、遺伝的アルゴリズムによって前記連結順序を決定することを特徴とする請求項 1 0 記載のコンパイラドライバ装置。

【請求項 1 3】 複数の中間オブジェクトを連結し、実行可能オブジェクトを作成するプログラムを記録した記録媒体において、

前記複数の中間オブジェクトの連結順序を決定する中間オブジェクト連結順序作成ステップと、

前記中間オブジェクト連結順序作成ステップにおいて決定された連結順序に基づき前記複数の中間オブジェクトを連結処理して実行可能オブジェクトを得るリンク処理ステップと、

前記リンク処理ステップによって得られた前記実行可能オブジェクトのプログラムサイズを前記連結順序が変更される毎に直前の前記連結順序の前記プログラムサイズと比較する比較ステップと、

前記比較ステップによる比較の結果、前記プログラムサイズの小さい方の前記プログラムサイズとその時の連結順序とを更新して記憶する記憶ステップと、

前記中間オブジェクト連結順序作成ステップにより前記連結順序を変更し、前記リンク処理ステップと前記比較ステップと前記記憶ステップとを繰り返し実行する繰り返しステップとをコンピュータに実行させるためのプログラムを記録した記録媒体。

【請求項 1 4】 前記中間オブジェクト連結順序作成ステップは、順列アルゴリズムによって前記連結順序を決定するステップであることを特徴とする請求項 1 3 記載の記録媒体。

【請求項 1 5】 前記中間オブジェクト連結順序作成ステップは、遺伝的アルゴリズムによって前記連結順序を決定するステップであることを特徴とする請求項 1 3 記載の記録媒体。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、中間オブジェクトを連結し、実行可能オブジェクトを作成する中間

オブジェクト連結方法、及び、中間オブジェクト連結装置、及び、リンカ装置、及び、コンパイラドライバ装置、並びに、中間オブジェクトを連結するプログラムを記録した記憶媒体技術に関する。

# 【 0 0 0 2 】

## 【従来の技術】

従来の中間オブジェクトの連結方法について、図 6 を参照して説明する。

図 6 は、従来の中間オブジェクトの連結方法に関するフローチャートである。先ず、連結する対象である中間オブジェクトのリストを入力する（ステップ 5 0）。次に、ステップ 5 0 で入力された中間オブジェクトのリストの順序に従い、中間オブジェクトを連結する等リンク処理を行い、実行可能オブジェクトを生成する。

そして、メモリアクセスに対するアドレッシングを複数持つ CPU は、作成された実行可能オブジェクトにおいて、分岐距離に応じた複数の分岐命令を保持する。

ここで、分岐命令のプログラムサイズは、分岐距離によって異なり、具体的には、分岐距離が長くなると分岐命令のプログラムサイズが大きくなる。更に、この分岐距離は、中間オブジェクトを連結する際の連結順序により変化するので、この結合順序によって、分岐命令を含む実行可能オブジェクトのプログラムサイズが変化する場合がある。

一般にプログラムサイズは、小さい方が好ましいので、設計者は、中間オブジェクトの結合順序を多数用意し、順次中間オブジェクトを連結する等リンク処理を行って、実行可能オブジェクトを作成し、実行可能オブジェクトのプログラムサイズが最小となる中間オブジェクトの連結順序を探し出す必要があった。

# 【 0 0 0 3 】

## 【発明が解決しようとする課題】

従来の方法において、プログラムサイズが最小となる中間オブジェクトの連結順序を探し出すには、いくつかの問題点があった。

先ず、最初の問題点は、最小サイズの実行可能オブジェクトを得るための中間オブジェクト連結順序の検討において、人手を介してしまうので手間がかかるこ

とである。

更に次の問題点は、中間オブジェクトの連結順序の組合せが多数にのぼるため、最小サイズの実行可能オブジェクトを作成する中間オブジェクトの連結順序を決定することは時間を有し、大変困難である。中間オブジェクトの個数がN個であった場合は、中間オブジェクトの連結順序の総数は、N！（階乗）個にもなってしまう。

#### 【 0 0 0 4 】

本発明はこのような状況に鑑みてなされたもので、最小サイズの実行可能オブジェクトを容易に作成することのできる中間オブジェクト連結方法、及び、中間オブジェクト連結装置、及び、リンカ装置、及び、コンパイラドライバ装置、並びに、中間オブジェクトを連結するプログラムを記録した記憶媒体を提供することを目的とする。

#### 【 0 0 0 5 】

##### 【課題を解決するための手段】

上記目的を達成するための第1の中間オブジェクト連結方法は、複数の中間オブジェクトを連結し、実行可能オブジェクトを作成する中間オブジェクト連結方法において、

前記複数の中間オブジェクトの連結順序を決定する中間オブジェクト連結順序作成ステップと、

前記中間オブジェクト連結順序作成ステップにおいて決定された連結順序に基づき前記複数の中間オブジェクトを連結処理して実行可能オブジェクトを得るリンク処理ステップと、

前記リンク処理ステップによって得られた前記実行可能オブジェクトのプログラムサイズを前記連結順序が変更される毎に直前の前記連結順序の前記プログラムサイズと比較する比較ステップと、

前記比較ステップによる比較の結果、前記プログラムサイズの小さい方の前記プログラムサイズとその時の連結順序とを更新して記憶する記憶ステップと、

前記中間オブジェクト連結順序作成ステップにより前記連結順序を変更し、前記リンク処理ステップと前記比較ステップと前記記憶ステップとを繰り返し実行

する繰り返しステップとを備えたことを特徴とする。

【 0 0 0 6 】

この中間オブジェクト連結方法によれば、中間オブジェクトの連結順序の組合せが多数であっても、記憶ステップにおいてプログラムサイズが最小時の連結順序を記憶するので、プログラムサイズが最小となる中間オブジェクトの連結順序を容易に得ることができる。

【 0 0 0 7 】

第 2 の中間オブジェクト連結方法では、前記中間オブジェクト連結順序作成ステップは、順列アルゴリズムによって前記連結順序を決定することを特徴とする。

【 0 0 0 8 】

この中間オブジェクト連結方法によれば、中間オブジェクトの全ての連結順序を順列により効率よく作成できるので、プログラムサイズが最小となる中間オブジェクトの連結順序を容易に得ることができる。

【 0 0 0 9 】

第 3 の中間オブジェクト連結方法では、前記中間オブジェクト連結順序作成ステップは、遺伝的アルゴリズムによって前記連結順序を決定することを特徴とする。

【 0 0 1 0 】

この中間オブジェクト連結方法によれば、遺伝的アルゴリズム適用することによって、中間オブジェクトの連結順序が多数であっても、効率的、且つ、高速に、プログラムサイズが最小となる中間オブジェクトの連結順序を得ることができる。

【 0 0 1 1 】

第 1 の中間オブジェクト連結装置では、複数の中間オブジェクトを連結し、実行可能オブジェクトを作成する中間オブジェクト連結装置において、

前記複数の中間オブジェクトの連結順序を決定する中間オブジェクト連結順序作成手段と、

前記中間オブジェクト連結順序作成手段において決定された連結順序に基づき

前記複数の中間オブジェクトを連結処理して実行可能オブジェクトを作成するリンク起動手段と、

前記リンク起動手段によって作成された前記実行可能オブジェクトのプログラムサイズを前記連結順序が変更される毎に直前の前記連結順序の前記プログラムサイズと比較する比較手段と、

前記比較手段による比較の結果、前記プログラムサイズの小さい方の前記プログラムサイズとその時の連結順序とを更新して記憶する記憶手段と、

前記中間オブジェクト連結順序作成手段により前記連結順序を変更し、前記リンク起動手段と前記比較手段と前記記憶手段とを繰り返し実行する繰り返し手段とを備えたことを特徴とする。

【 0 0 1 2 】

この中間オブジェクト連結装置によれば、最小サイズの実行可能オブジェクトを得るための中間オブジェクト連結順序の検討において、人手を介すことがないので手間がかからない。

また、中間オブジェクトの連結順序の組合せが多数であっても、記憶手段によってプログラムサイズが最小時の連結順序を記憶するので、プログラムサイズが最小となる中間オブジェクトの連結順序を容易に得ることができる。

【 0 0 1 3 】

第2の中間オブジェクト連結装置では、前記中間オブジェクト連結順序作成手段は、順列アルゴリズムによって前記連結順序を決定することを特徴とする。

【 0 0 1 4 】

この中間オブジェクト連結装置によれば、中間オブジェクトの全ての連結順序を順列により効率よく作成できるので、プログラムサイズが最小となる中間オブジェクトの連結順序を容易に得ることができる。

【 0 0 1 5 】

第3の中間オブジェクト連結装置では、前記中間オブジェクト連結順序作成手段は、遺伝的アルゴリズムによって前記連結順序を決定することを特徴とする。

【 0 0 1 6 】

この中間オブジェクト連結装置によれば、遺伝的アルゴリズム適用することに

よって、中間オブジェクトの連結順序が多数であっても、効率的、且つ、高速に、プログラムサイズが最小となる中間オブジェクトの連結順序を得ることができる。

【 0 0 1 7 】

第 1 のリンカ装置では、複数の中間オブジェクトを連結し、実行可能オブジェクトを作成するリンカ装置において、

前記複数の中間オブジェクトの連結順序を決定する中間オブジェクト連結順序作成手段と、

前記中間オブジェクト連結順序作成手段において決定された連結順序に基づき前記複数の中間オブジェクトを連結処理して実行可能オブジェクトを作成するリンク処理手段と、

前記リンク処理手段によって作成された前記実行可能オブジェクトのプログラムサイズを前記連結順序が変更される毎に直前の前記連結順序の前記プログラムサイズと比較する比較手段と、

前記比較手段による比較の結果、前記プログラムサイズの小さい方の前記プログラムサイズとその時の連結順序とを更新して記憶する記憶手段と、

前記中間オブジェクト連結順序作成手段により前記連結順序を変更し、前記リンク処理手段と前記比較手段と前記記憶手段とを繰り返し実行する繰り返し手段とを備えたことを特徴とする。

【 0 0 1 8 】

このリンカ装置によれば、最小サイズの実行可能オブジェクトを得るための中間オブジェクト連結順序の検討において、人手を介さないので手間がかからない。

また、中間オブジェクトの連結順序の組合せが多数であっても、記憶手段によってプログラムサイズが最小時の連結順序を記憶するので、プログラムサイズが最小となる中間オブジェクトの連結順序を容易に得ることができる。

【 0 0 1 9 】

第 2 のリンカ装置では、前記中間オブジェクト連結順序作成手段は、順列アルゴリズムによって前記連結順序を決定することを特徴とする。

【 0 0 2 0 】

このリンカ装置によれば、中間オブジェクトの全ての連結順序を順列により効率よく作成できるので、プログラムサイズが最小となる中間オブジェクトの連結順序を容易に得ることができる。

【 0 0 2 1 】

第 3 のリンカ装置では、前記中間オブジェクト連結順序作成手段は、遺伝的アルゴリズムによって前記連結順序を決定することを特徴とする。

【 0 0 2 2 】

このリンカ装置によれば、遺伝的アルゴリズム適用することによって、中間オブジェクトの連結順序が多数であっても、効率的、且つ、高速に、プログラムサイズが最小となる中間オブジェクトの連結順序を得ることができる。

【 0 0 2 3 】

第 1 のコンパイラドライバ装置では、コンパイラ、アセンブラ、リンカ等を起動することによって、ソースプログラムを翻訳し実行可能オブジェクトの作成を行うコンパイラドライバ装置において、

前記複数の中間オブジェクトの連結順序を決定する中間オブジェクト連結順序作成手段と、

前記中間オブジェクト連結順序作成手段において決定された連結順序に基づき前記複数の中間オブジェクトを連結処理して実行可能オブジェクトを作成するリンカ起動手段と、

前記リンカ起動手段によって作成された前記実行可能オブジェクトのプログラムサイズを前記連結順序が変更される毎に直前の前記連結順序の前記プログラムサイズと比較する比較手段と、

前記比較手段による比較の結果、前記プログラムサイズの小さい方の前記プログラムサイズとその時の連結順序とを更新して記憶する記憶手段と、

前記中間オブジェクト連結順序作成手段により前記連結順序を変更し、前記リンカ起動手段と前記比較手段と前記記憶手段とを繰り返し実行する繰り返し手段とを備えたことを特徴とする。

【 0 0 2 4 】

このコンパイラドライバ装置によれば、最小サイズの実行可能オブジェクトを得るための中間オブジェクト連結順序の検討において、人手を介さないので手間がかからない。

また、中間オブジェクトの連結順序の組合せが多数であっても、記憶手段によってプログラムサイズが最小時の連結順序を記憶するので、プログラムサイズが最小となる中間オブジェクトの連結順序を容易に得ることができる。

【 0 0 2 5 】

第 2 のコンパイラドライバ装置では、前記中間オブジェクト連結順序作成手段は、順列アルゴリズムによって前記連結順序を決定することを特徴とする。

【 0 0 2 6 】

このコンパイラドライバ装置によれば、中間オブジェクトの全ての連結順序を順列により効率よく作成できるので、プログラムサイズが最小となる中間オブジェクトの連結順序を容易に得ることができる。

【 0 0 2 7 】

第 3 のコンパイラドライバ装置は、前記中間オブジェクト連結順序作成手段は、遺伝的アルゴリズムによって前記連結順序を決定することを特徴とする。

【 0 0 2 8 】

このコンパイラドライバ装置によれば、遺伝的アルゴリズム適用することによって、中間オブジェクトの連結順序が多数であっても、効率的、且つ、高速に、プログラムサイズが最小となる中間オブジェクトの連結順序を得ることができる。

【 0 0 2 9 】

第 1 の記録媒体は、複数の中間オブジェクトを連結し、実行可能オブジェクトを作成するプログラムを記録した記録媒体において、

前記複数の中間オブジェクトの連結順序を決定する中間オブジェクト連結順序作成ステップと、

前記中間オブジェクト連結順序作成ステップにおいて決定された連結順序に基づき前記複数の中間オブジェクトを連結処理して実行可能オブジェクトを得るリンク処理ステップと、



前記リンク処理ステップによって得られた前記実行可能オブジェクトのプログラムサイズを前記連結順序が変更される毎に直前の前記連結順序の前記プログラムサイズと比較する比較ステップと、

前記比較ステップによる比較の結果、前記プログラムサイズの小さい方の前記プログラムサイズとその時の連結順序とを更新して記憶する記憶ステップと、

前記中間オブジェクト連結順序作成ステップにより前記連結順序を変更し、前記リンク処理ステップと前記比較ステップと前記記憶ステップとを繰り返し実行する繰り返しステップとをコンピュータに実行させるためのプログラムを記録したことを特徴とする。

【 0 0 3 0 】

この記録媒体によれば、最小サイズの実行可能オブジェクトを得るための中間オブジェクト連結順序の検討をプログラムによって行うので、人手による手間がかからない。

また、中間オブジェクトの連結順序の組合せが多数であっても、記憶ステップによってプログラムサイズが最小時の連結順序を記憶するので、プログラムサイズが最小となる中間オブジェクトの連結順序を容易に得ることができる

【 0 0 3 1 】

第 2 の記録媒体は、前記中間オブジェクト連結順序作成ステップは、順列アルゴリズムによって前記連結順序を決定するステップであることを特徴とする。

【 0 0 3 2 】

この記録媒体によれば、中間オブジェクトの全ての連結順序を順列により効率よく作成できるので、プログラムサイズが最小となる中間オブジェクトの連結順序を容易に得ることができる。

【 0 0 3 3 】

第 3 の記録媒体は、前記中間オブジェクト連結順序作成ステップは、遺伝的アルゴリズムによって前記連結順序を決定するステップであることを特徴とする。

【 0 0 3 4 】

この記録媒体によれば、遺伝的アルゴリズム適用することによって、中間オブジェクトの連結順序が多数であっても、効率的、且つ、高速に、プログラムサイ

ズが最小となる中間オブジェクトの連結順序を得ることができる。

【 0 0 3 5 】

【発明の実施の形態】

〔第 1 実施形態〕

本実施形態の中間オブジェクト連結装置と周辺機器との構成について、図 1 を参照して説明する。

図 1 は、第 1 実施形態の中間オブジェクト連結装置と周辺機器との構成を示す構成図である。

中間オブジェクト連結装置 1 0 は、中間オブジェクト連結順序作成手段である連結順序作成部 1 1 と、リンカ起動手段であるリンカ起動部 1 2 と、比較手段である比較部 1 3 と、記憶手段である記憶部 1 4 と、繰り返し手段である繰り返し部 1 5 とから構成されている。

【 0 0 3 6 】

連結順序作成部 1 1 は、中間オブジェクト 1 6 の連結順序を作成するユニットであり、連結順序作成部 1 1 で作成された連結順序は、連結順序部 1 1 a によって、次の連結順序が作成されるまで保持される。この連結順序部 1 1 a に保持された連結順序に基づいて、リンカ起動部 1 2 が、リンカ 1 7 を起動し、リンク処理を行わせることにより、実行可能オブジェクト 1 8 が作成される。

また、連結順序作成部 1 1 は、後述する順列アルゴリズムや遺伝的アルゴリズム等によって、連結順序を作成する。

リンカ起動部 1 2 は、連結順序作成部 1 1 で作成された連結順序に基づいてリンカ 1 7 を起動し、リンク処理を行わせ、実行可能オブジェクト 1 8 が作成される。

比較部 1 3 は、リンカ 1 7 によって作成された実行可能オブジェクト 1 8 のプログラムサイズと、記憶部 1 4 に記憶されている実行可能オブジェクトのプログラムサイズとを比較する。

【 0 0 3 7 】

記憶部 1 4 は、比較部 1 3 の比較によって、プログラムサイズが小さい方のプログラムサイズを記憶して更新する。そして、後述する繰り返し部 1 5 による各

工程の繰り返しにより、作成された実行可能オブジェクト 1 8 のうち最小のプログラムサイズを記憶することになる。また、記憶部 1 4 は、小さい方のプログラムサイズを更新して記録する際に、連結順序部 1 1 a から、そのプログラムサイズの実行可能オブジェクト 1 8 を得た時の連結順序も記憶する。

よって、記憶部 1 4 には、最小のプログラムサイズと、この最小のプログラムサイズの実行可能オブジェクト 1 8 を得た時の中間オブジェクト 1 6 の連結順序とを記憶している。

また、記憶部 1 4 の初期段階においては、プログラムサイズの初期値を予め入力しておくといよい。例えば、初期値としては、プログラムサイズとしてあり得ない大きさの値を入力しておけばよい。

#### 【 0 0 3 8 】

繰り返し部 1 5 は、比較部 1 3 での比較後に、再度、連結順序作成部 1 1 に工程を戻すことによって、連結順序作成部 1 1 からリンカ起動部 1 2、そして比較部 1 3 と記憶部 1 4 へと一連の工程を繰り返させるためのユニットである。

以上が、中間オブジェクト連結装置 1 0 の内部構成である。

#### 【 0 0 3 9 】

中間オブジェクト連結装置 1 0 の周辺機器には、中間オブジェクト 1 6 と、リンカ 1 7 と、実行可能オブジェクト 1 8 が存在する。

実行可能オブジェクト 1 8 は、分岐命令を含み、この分岐命令は、分岐距離によってプログラムサイズが異なる。また、この分岐距離が長くなると、実行可能オブジェクトのプログラムサイズが大きくなる。

また、分岐距離は、中間オブジェクト 1 6 の連結順序によって変化するので、中間オブジェクト 1 6 の連結順序により、実行可能オブジェクト 1 8 のプログラムサイズが変化する。

以上が、中間オブジェクト連結装置 1 0 の周辺機器の構成である。そして、中間オブジェクト連結装置 1 0 に順列アルゴリズムを適用した場合と、遺伝的アルゴリズムを適用した場合を以下に説明する。

#### 【 0 0 4 0 】

##### ( 1 ) 順列アルゴリズムを適用した場合

次に、上記の中間オブジェクト連結装置 1 0 に順列アルゴリズムを適用した場合の動作について図 2 を参照して説明する。

図 2 は、中間オブジェクト連結装置に順列アルゴリズムを適用した時のフローチャートである。

#### 【 0 0 4 1 】

まず、記憶部 1 4 に記憶されるプログラムサイズの最小値（初期値）を入力し（ステップ 1）、中間オブジェクト 1 6 のリストを入力する（ステップ 2）。尚、この初期値やリストの入力は、図示しない入力装置によって入力される。

そして、連結順序作成部 1 1 が最初に作成した連結順序に基づいて、リンカ 1 7 は、中間オブジェクト 1 6 を連結する等リンク処理を行って、実行可能オブジェクト 8 を得る（ステップ 3）。

#### 【 0 0 4 2 】

次に、比較部 1 3 によって、ステップ 3 において作成された実行可能オブジェクト 1 8 のプログラムサイズと、記憶部 1 4 に記憶されているプログラムサイズの最小値（初期値）とを比較する（ステップ 4）。プログラムサイズが最小値よりも小さくない時は、ステップ 7 に進む。

最小値（初期値）は、十分大きな値であり、プログラムサイズは最小値（初期値）よりも小さいので、ステップ 5 に進む。

プログラムサイズは最小値（初期値）よりも小さいので、記憶部 1 4 にこのプログラムサイズが最小値として記録されて更新される（ステップ 5）。そして、この記録されたプログラムサイズに対する連結順序を連結順序部 1 1 a から読み出して、この時の連結順序を記憶部 1 4 に記録する（ステップ 6）。

#### 【 0 0 4 3 】

次に、繰り返し部 1 5 によって、連結順序作成部 1 1 に戻されて、順列アルゴリズムに基づき、次の連結順序を作成（更新）する（ステップ 7）。

ここで、順列アルゴリズムとは、順列で求められる全ての並べ方を順次求めたり、順次切り換えたりするアルゴリズムである。例えば、中間オブジェクト 1 6 の連結順序を決定することに対して、順列アルゴリズムを適用することを考える。中間オブジェクト 1 6 が N 個あり、全ての中間オブジェクト 1 6 を連結させる

連結順序を求める場合では、N個の中間オブジェクト16の連結順序を順列とすると、N！（階乗）通りの連結順序が求められる。

【0044】

フローチャートに説明を戻す。

ステップ7の後は、ステップ3に戻し（ステップ8）、リンク処理を行い、実行可能オブジェクトを作成し、ステップ3～ステップ7を繰り返す。このステップ8では、連結順序作成部11によって連結順序が全て作成されたか否かの判断を行っている。

ステップ8において、連結順序が全て作成されて終了した場合に、記憶部14には、全ての連結順序によって作成された実行可能オブジェクト18のうち、最小のプログラムサイズと、その時（最小時）の連結順序が記憶されているので、この最小時の連結順序を用いて、中間オブジェクト16をリンク処理し、プログラムサイズが最小となる実行可能オブジェクト18を作成することができる（ステップ9）。

【0045】

よって、この中間オブジェクト連結装置10によれば、最小サイズの実行可能オブジェクト18を得るための中間オブジェクト16の連結順序検討において、人手を介すことがないので手間がかからない。

また、中間オブジェクト16の連結順序の組合せが多数であっても、中間オブジェクト16の全ての連結順序を順列により効率よく作成でき、記憶部14によってプログラムサイズが最小時の連結順序を記憶するので、プログラムサイズが最小となる中間オブジェクト16の連結順序を容易に得ることができる。

【0046】

（2）遺伝的アルゴリズムを適用した場合

次に、上記の中間オブジェクト連結装置10に遺伝的アルゴリズムを適用した場合の動作について図3を参照して説明する。

図3は、中間オブジェクト連結装置に遺伝的アルゴリズムを適用した時のフローチャートである。

【0047】

ここで、遺伝的アルゴリズム (Genetic Algorithm) とは、生物の染色体やゲノムによる遺伝子モデルを応用したアルゴリズムである。生物の遺伝では、ある世代の遺伝子が、次の世代に遺伝する際には、淘汰、交叉、突然変異等の遺伝操作を行う。また、遺伝子は、その遺伝モデルに基づく評価関数によって評価され、ある評価値を持っている。そして、遺伝子は、この評価値が高くなるように遺伝子操作を行って、より高い評価値を得ようとする。

また、評価値が高くなる遺伝子操作方法、即ち、評価値が高くなる為の淘汰、交叉、突然変異の方法は、この遺伝モデルにおいて高い評価値を得る最適解といえる。

よって、遺伝的アルゴリズムは、ある大規模で多岐に渡る変数を持つモデルにおいて、この変数の最適解を得るのに適したアルゴリズムである。

#### 【 0 0 4 8 】

遺伝的アルゴリズムを工業的なモデルに適用する場合は、例えば、遺伝子は、文字列 (数列含む) で表される。また、淘汰は、評価値の低い文字列の削除と評価値の高い文字列の採用であり、交叉は、対称となる遺伝子の一部文字列の交換であり、突然変異は、文字列の一部を置き換えることである。

よって、本実施例における遺伝子は、中間オブジェクト 1 6 の結合順序とみなすことができ、実行可能オブジェクト 1 8 のプログラムサイズが小さくなる程、評価値が高くなるとすることができる。

#### 【 0 0 4 9 】

図 3 のフローチャートを説明する。まず、記憶部 1 4 に記憶されるプログラムサイズの最小値 (初期値) を入力し (ステップ 1 0)、中間オブジェクト 1 6 のリストを入力する (ステップ 1 1)。尚、この初期値やリストの入力は、図示しない入力装置によって入力される。

そして、ステップ 1 1 で入力されたリストにより、所定個数の遺伝子 (結合順序を持つ) を作成しする (遺伝子の初期化) (ステップ 1 2)。

それから、連結順序作成部 1 1 が最初に作成した所定個数の一つの遺伝子に基づいて、リンカ 1 7 は、中間オブジェクト 1 6 を連結する等リンク処理を行って、実行可能オブジェクト 8 を得る (ステップ 1 3)。

## 【 0 0 5 0 】

次に、比較部 1 3 によって、ステップ 1 3 において作成された実行可能オブジェクト 1 8 のプログラムサイズと、記憶部 1 4 に記憶されているプログラムサイズの最小値（初期値）とを比較する（ステップ 1 4）。プログラムサイズが最小値よりも小さくない時は、ステップ 1 7 に進む。

最小値（初期値）は、十分大きな値であり、プログラムサイズは最小値（初期値）よりも小さいので、ステップ 1 5 に進む。

プログラムサイズは最小値（初期値）よりも小さいので、記憶部 1 4 にこのプログラムサイズが最小値として記録されて更新される（ステップ 1 5）。そして、この記録されたプログラムサイズに対する連結順序を連結順序部 1 1 a から読み出して、この時の連結順序を記憶部 1 4 に記録する（ステップ 1 6）。

## 【 0 0 5 1 】

次に、繰り返し部 1 5 によって、連結順序作成部 1 1 に戻されて、作成された所定個数のうち、他の遺伝子について、ステップ 1 3 ～ステップ 1 6 を繰り返す（ステップ 1 7）。

遺伝子の所定個数だけ繰り返した後に、ステップ 1 3 のリンク処理によって作成された実行可能オブジェクトのプログラムサイズに基づいて、遺伝子の交叉を行う（ステップ 1 8）。例えば、プログラムサイズが所定の閾値より大きかったら、より小さいプログラムサイズになるように遺伝子の交叉を行う。

次に、ステップ 1 8 と同様に、ステップ 1 3 のリンク処理によって求められたプログラムサイズに基づいて、遺伝子の突然変異を行う（ステップ 1 9）。例えば、プログラムサイズが所定の閾値より大きかったら、より小さいプログラムサイズになるように遺伝子の突然変異を行う。

尚、ステップ 1 8、1 9 の順番は逆でも良いし、閾値も個別に設定しても良い。

## 【 0 0 5 2 】

ステップ 1 9 の後は、ステップ 1 3 に戻し（ステップ 2 0）、リンク処理を行い、実行可能オブジェクトを作成し、ステップ 1 3 ～ステップ 1 9 を繰り返す。このステップ 2 0 では、連結順序作成部 1 1 によって連結順序の作成を終了する

か否かの判定を行っており、この判定は、ステップ 1 3 に所定回数戻したか、或いは、ステップ 1 3 のリンク処理によって求められたプログラムサイズが所定の値より小さくなったかによって終了の判定を行い、次のステップに進む。

ステップ 2 0 において、終了判定した場合に、記憶部 1 4 には、作成された実行可能オブジェクト 1 8 のうち、最小のプログラムサイズと、その時（最小時）の連結順序が記憶されているので、この最小時の連結順序を用いて、中間オブジェクト 1 6 をリンク処理し、プログラムサイズが最小となる実行可能オブジェクト 1 8 を作成することができる（ステップ 2 1）。

#### 【 0 0 5 3 】

よって、最小サイズの実行可能オブジェクト 1 8 を得るための中間オブジェクト 1 6 の連結順序検討において、人手を介すことがないので手間がかからない。

また、中間オブジェクト 1 6 の連結順序の組合せが多数であっても、遺伝的アルゴリズム適用することによって、効率的、且つ、高速に、プログラムサイズが最小となる中間オブジェクト 1 6 の連結順序を容易に得ることができる。

#### 【 0 0 5 4 】

##### 〔第 2 の実施形態〕

本実施形態のリンカ装置と周辺機器との構成について、図 4 を参照して説明する。

図 4 は、第 2 実施形態のリンカ装置と周辺機器との構成を示す構成図である。

リンカ装置 3 0 は、中間オブジェクト連結順序作成手段である連結順序作成部 3 1 と、リンカ処理手段であるリンカ処理部 3 2 と、比較手段である比較部 3 3 と、記憶手段である記憶部 3 4 と、繰り返し手段である繰り返し部 3 5 とから構成されている。

#### 【 0 0 5 5 】

連結順序作成部 3 1 は、中間オブジェクト 3 6 の連結順序を作成するユニットであり、連結順序作成部 3 1 で作成された連結順序は、連結順序部 3 1 a によって、次の連結順序が作成されるまで保持される。この連結順序部 3 1 a に保持された連結順序に基づいて、リンカ処理部 3 2 が、リンク処理を行うことにより、実行可能オブジェクト 3 8 が作成される。



また、連結順序作成部 3 1 は、第 1 の実施形態に記載した順列アルゴリズムや遺伝的アルゴリズムと同様に、連結順序を作成する。

リンカ処理部 3 2 は、連結順序作成部 3 1 で作成された連結順序に基づいてリンク処理し、実行可能オブジェクト 3 8 を作成する。

比較部 3 3 は、リンカ処理部 3 2 によって作成された実行可能オブジェクト 3 8 のプログラムサイズと、記憶部 3 4 に記憶されている実行可能オブジェクトのプログラムサイズとを比較する。

【 0 0 5 6 】

記憶部 3 4 は、比較部 3 3 の比較によって、プログラムサイズが小さい方のプログラムサイズを記憶して更新する。そして、後述する繰り返し部 3 5 による各工程の繰り返しにより、作成された実行可能オブジェクト 3 8 のうち最小のプログラムサイズを記憶することになる。また、記憶部 3 4 は、小さい方のプログラムサイズを更新して記録する際に、連結順序部 3 1 a から、そのプログラムサイズの実行可能オブジェクト 3 8 を得た時の連結順序も記憶する。

よって、記憶部 3 4 には、最小のプログラムサイズと、この最小のプログラムサイズの実行可能オブジェクト 3 8 を得た時の中間オブジェクト 3 6 の連結順序とを記憶している。

また、記憶部 3 4 の初期段階においては、プログラムサイズの初期値を予め入力しておくといよい。例えば、初期値としては、プログラムサイズとしてあり得ない大きさの値を入力しておけばよい。

【 0 0 5 7 】

繰り返し部 3 5 は、比較部 3 3 での比較後に、再度、連結順序作成部 3 1 に工程を戻すことによって、連結順序作成部 3 1 からリンカ処理部 3 2、そして比較部 3 3 と記憶部 3 4 へと一連の工程を繰り返させるためのユニットである。

以上が、リンカ装置 3 0 の内部構成である。

【 0 0 5 8 】

リンカ装置 3 0 の周辺機器には、中間オブジェクト 3 6 と、実行可能オブジェクト 3 8 が存在する。

実行可能オブジェクト 3 8 は、分岐命令を含み、この分岐命令は、分岐距離に

よってプログラムサイズが異なる。また、この分岐距離が長くなると、実行可能オブジェクトのプログラムサイズが大きくなる。

また、分岐距離は、中間オブジェクト 3 6 の連結順序によって変化するので、中間オブジェクト 3 6 の連結順序により、実行可能オブジェクト 3 8 のプログラムサイズが変化する。

以上が、リンカ装置 3 0 の周辺機器の構成である。

尚、リンカ装置 3 0 に順列アルゴリズムを適用した場合と、遺伝的アルゴリズムを適用した場合の動作は、第 1 の実施形態と同様であるので説明を省略する。

#### 【 0 0 5 9 】

よって、このリンカ装置 3 0 によれば、最小サイズの実行可能オブジェクト 3 8 を得るための中間オブジェクト 3 6 の連結順序検討において、人手を介すことがないので手間がかからない。

また、中間オブジェクト 3 6 の連結順序の組合せが多数であっても、中間オブジェクト 3 6 の全ての連結順序を順列アルゴリズムにより効率よく作成でき、記憶部 3 4 によってプログラムサイズが最小時の連結順序を記憶するので、プログラムサイズが最小となる中間オブジェクト 3 6 の連結順序を容易に得ることができる。

また、中間オブジェクト 3 6 の連結順序の組合せが多数であっても、遺伝的アルゴリズム適用することによって、効率的、且つ、高速に、プログラムサイズが最小となる中間オブジェクト 3 6 の連結順序を容易に得ることができる。

#### 【 0 0 6 0 】

##### 〔第 3 の実施形態〕

本実施形態のコンパイラドライバ装置と周辺機器との構成について、図 5 を参照して説明する。

図 5 は、第 3 実施形態のコンパイラドライバ装置と周辺機器との構成を示す構成図である。

コンパイラドライバ装置 5 0 は、中間オブジェクト連結順序作成手段である連結順序作成部 5 1 と、リンカ起動手段であるリンカ起動部 5 2 と、比較手段である比較部 5 3 と、記憶手段である記憶部 5 4 と、繰り返し手段である繰り返し部

5 5 とから構成されている。

【 0 0 6 1 】

連結順序作成部 5 1 は、中間オブジェクト 5 6 の連結順序を作成するユニットであり、連結順序作成部 5 1 で作成された連結順序は、連結順序部 5 1 a によって、次の連結順序が作成されるまで保持される。この連結順序部 5 1 a に保持された連結順序に基づいて、リンカ起動部 5 2 が、リンカ 5 7 を起動し、リンク処理を行わせることにより、実行可能オブジェクト 5 8 が作成される。

また、連結順序作成部 5 1 は、第 1 の実施形態に記載した順列アルゴリズムや遺伝的アルゴリズムと同様に、連結順序を作成する。

リンカ起動部 5 2 は、連結順序作成部 5 1 で作成された連結順序に基づいてリンカ 5 7 を起動し、リンク処理を行わせ、実行可能オブジェクト 5 8 が作成される。

比較部 5 3 は、リンカ 1 7 によって作成された実行可能オブジェクト 5 8 のプログラムサイズと、記憶部 5 4 に記憶されている実行可能オブジェクトのプログラムサイズとを比較する。

【 0 0 6 2 】

記憶部 5 4 は、比較部 5 3 の比較によって、プログラムサイズが小さい方のプログラムサイズを記憶して更新する。そして、後述する繰り返し部 5 5 による各工程の繰り返しにより、作成された実行可能オブジェクト 5 8 のうち最小のプログラムサイズを記憶することになる。また、記憶部 5 4 は、小さい方のプログラムサイズを更新して記録する際に、連結順序部 5 1 a から、そのプログラムサイズの実行可能オブジェクト 5 8 を得た時の連結順序も記憶する。

よって、記憶部 5 4 には、最小のプログラムサイズと、この最小のプログラムサイズの実行可能オブジェクト 5 8 を得た時の中間オブジェクト 5 6 の連結順序とを記憶している。

また、記憶部 5 4 の初期段階においては、プログラムサイズの初期値を予め入力しておくといよい。例えば、初期値としては、プログラムサイズとしてあり得ない大きさの値を入力しておけばよい。

【 0 0 6 3 】

繰り返し部 5 5 は、比較部 5 3 での比較後に、再度、連結順序作成部 5 1 に工程を戻すことによって、連結順序作成部 5 1 からリンカ起動部 5 2、そして比較部 5 3 と記憶部 5 4 へと一連の工程を繰り返させるためのユニットである。

以上が、コンパイラドライバ装置 5 0 の内部構成である。

#### 【 0 0 6 4 】

コンパイラドライバ装置 5 0 の周辺機器には、中間オブジェクト 5 6 と、リンカ 5 7 と、実行可能オブジェクト 5 8 が存在する。

実行可能オブジェクト 5 8 は、分岐命令を含み、この分岐命令は、分岐距離によってプログラムサイズが異なる。また、この分岐距離が長くなると、実行可能オブジェクトのプログラムサイズが大きくなる。

また、分岐距離は、中間オブジェクト 5 6 の連結順序によって変化するので、中間オブジェクト 5 6 の連結順序により、実行可能オブジェクト 5 8 のプログラムサイズが変化する。

以上が、コンパイラドライバ装置 5 0 の周辺機器の構成である。

尚、コンパイラドライバ装置 5 0 に順列アルゴリズムを適用した場合と、遺伝的アルゴリズムを適用した場合の動作は、第 1 の実施形態と同様であるので説明を省略する。

#### 【 0 0 6 5 】

よって、このコンパイラドライバ装置 5 0 によれば、最小サイズの実行可能オブジェクト 5 8 を得るための中間オブジェクト 5 6 の連結順序検討において、人手を介することがないので手間がかからない。

また、中間オブジェクト 5 6 の連結順序の組合せが多数であっても、中間オブジェクト 5 6 の全ての連結順序を順列アルゴリズムにより効率よく作成でき、記憶部 5 4 によってプログラムサイズが最小時の連結順序を記憶するので、プログラムサイズが最小となる中間オブジェクト 5 6 の連結順序を容易に得ることができる。

また、中間オブジェクト 5 6 の連結順序の組合せが多数であっても、遺伝的アルゴリズム適用することによって、効率的、且つ、高速に、プログラムサイズが最小となる中間オブジェクト 5 6 の連結順序を容易に得ることができる。

【 0 0 6 6 】

【発明の効果】

本発明の中間オブジェクト連結装置によれば、最小サイズの実行可能オブジェクトを得るための中間オブジェクト連結順序の検討において、人手を介すことがないので手間がかからない。

また、中間オブジェクトの連結順序の組合せが多数であっても、記憶手段によってプログラムサイズが最小時の連結順序を記憶するので、プログラムサイズが最小となる中間オブジェクトの連結順序を容易に得ることができる。

また、中間オブジェクトの全ての連結順序を順列により効率よく作成できるので、プログラムサイズが最小となる中間オブジェクトの連結順序を容易に得ることができる。

また、遺伝的アルゴリズム適用することによって、中間オブジェクトの連結順序が多数であっても、効率的、且つ、高速に、プログラムサイズが最小となる中間オブジェクトの連結順序を得ることができる。

【図面の簡単な説明】

【図 1】

第 1 実施形態の中間オブジェクト連結装置と周辺機器との構成を示す構成図である。

【図 2】

中間オブジェクト連結装置に順列アルゴリズムを適用した時のフローチャートである。

【図 3】

中間オブジェクト連結装置に遺伝的アルゴリズムを適用した時のフローチャートである。

【図 4】

第 2 実施形態のリンカ装置と周辺機器との構成を示す構成図である。

【図 5】

第 3 実施形態のコンパイラドライバ装置と周辺機器との構成を示す構成図である。

【図 6】

図 6 は、従来の中間オブジェクトの連結方法に関するフローチャートである。

【符号の説明】

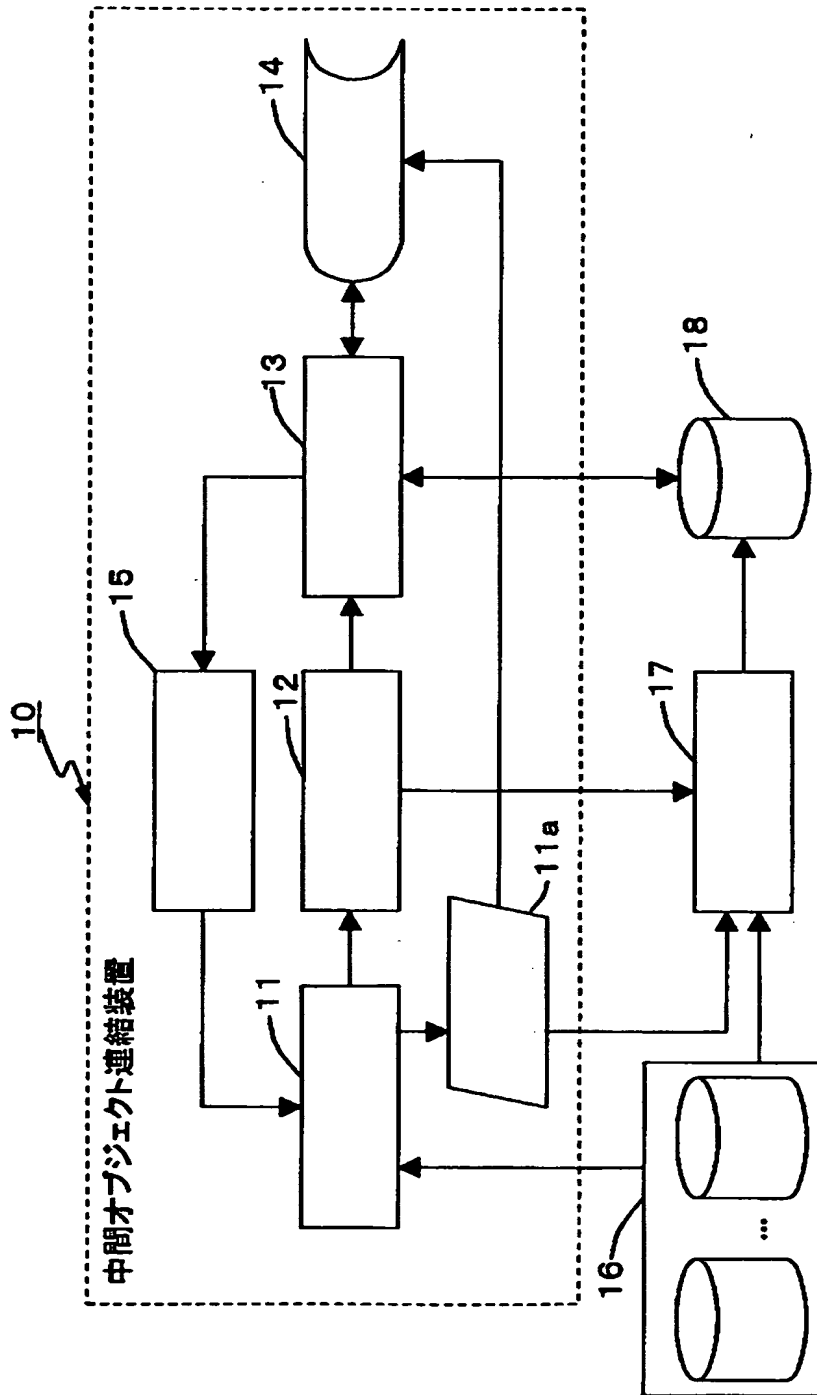
- 1 0      中間オブジェクト連結装置
- 1 1      連結順序作成部
- 1 1 a    連結順序部
- 1 2      リンカ起動部
- 1 3      比較部
- 1 4      記憶部
- 1 5      繰り返し部
- 1 6      中間オブジェクト
- 1 7      リンカ
- 1 8      実行可能オブジェクト
- 3 0      リンカ装置
- 3 1      連結順序作成部
- 3 1 a    連結順序部
- 3 2      リンカ処理部
- 3 3      比較部
- 3 4      記憶部
- 3 5      繰り返し部
- 3 6      中間オブジェクト
- 3 8      実行可能オブジェクト
- 5 0      コンパイラドライバ装置
- 5 1      連結順序作成部
- 5 1 a    連結順序部
- 5 2      リンカ起動部
- 5 3      比較部
- 5 4      記憶部

- 5 5 繰り返し部
- 5 6 中間オブジェクト
- 5 7 リンカ
- 5 8 実行可能オブジェクト

【書類名】

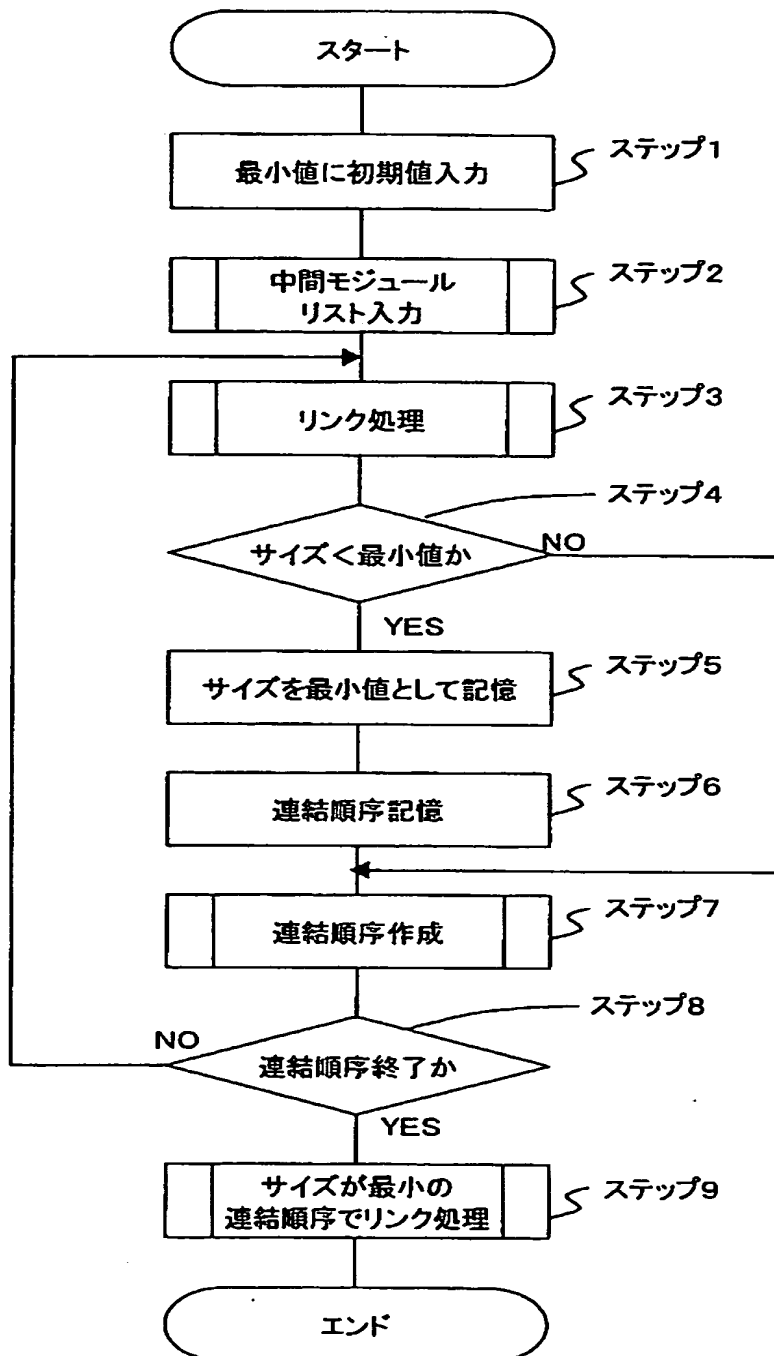
図面

【図 1】

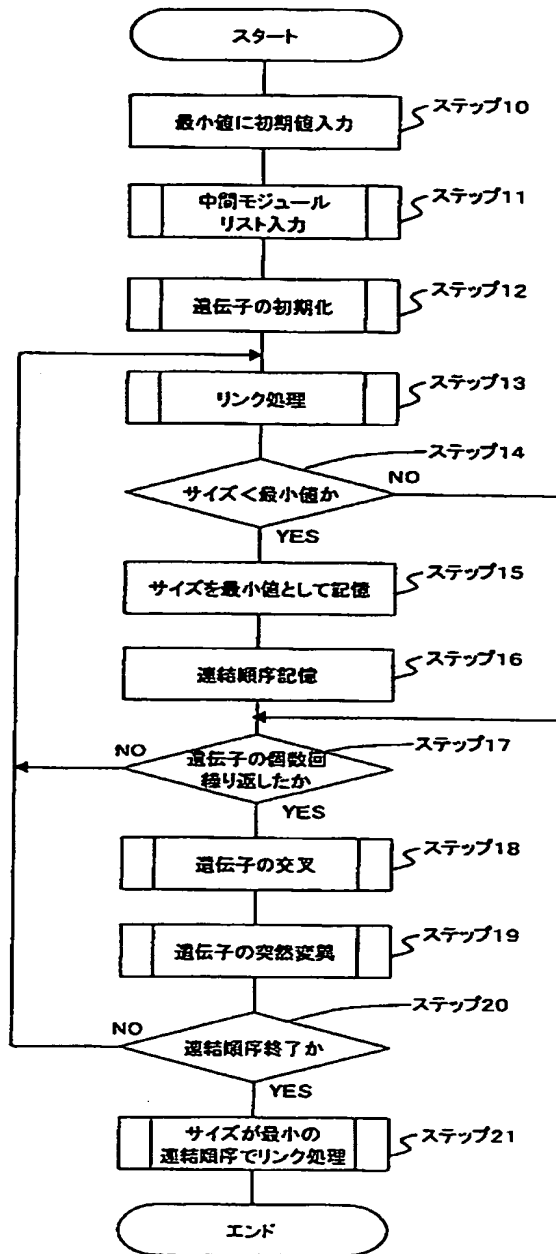




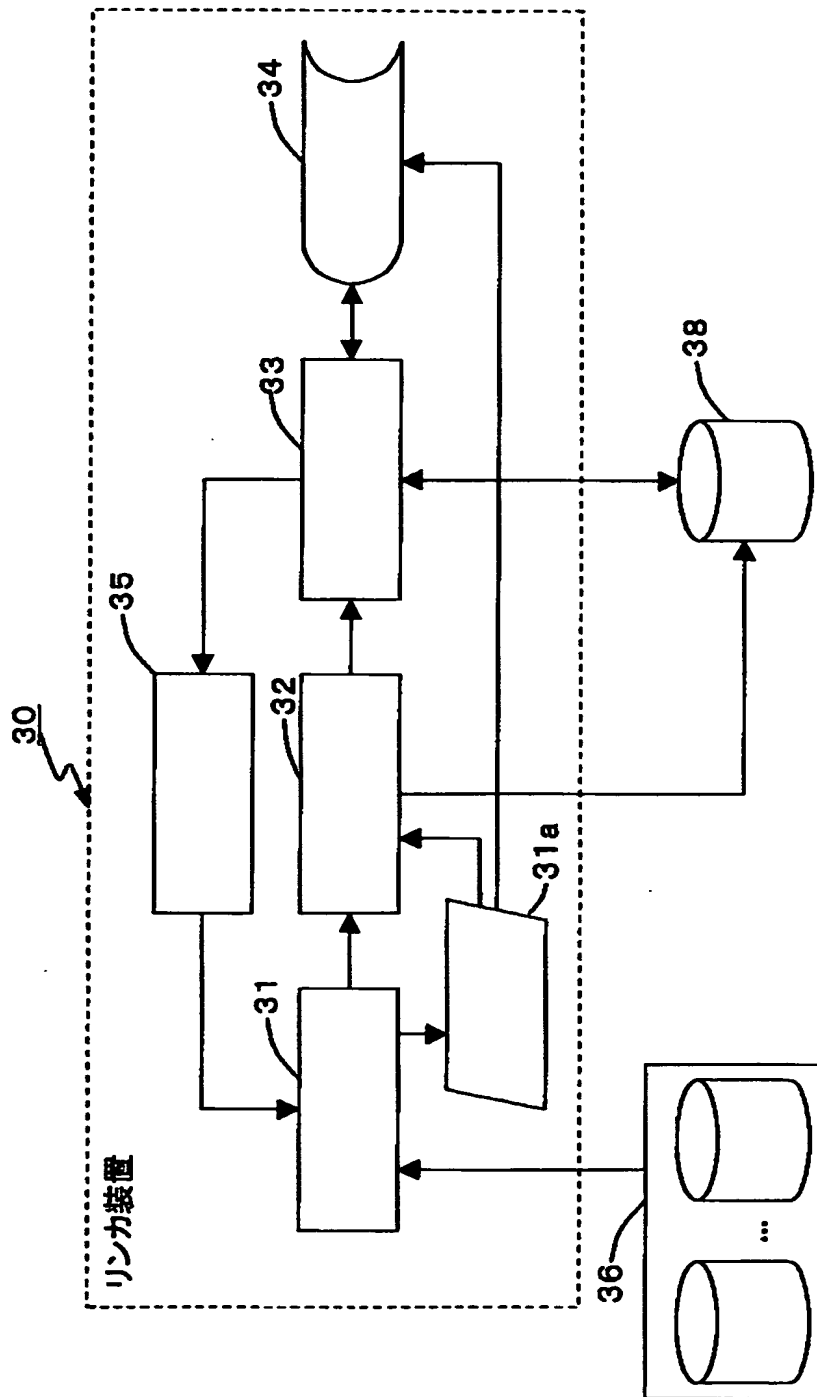
【図 2】



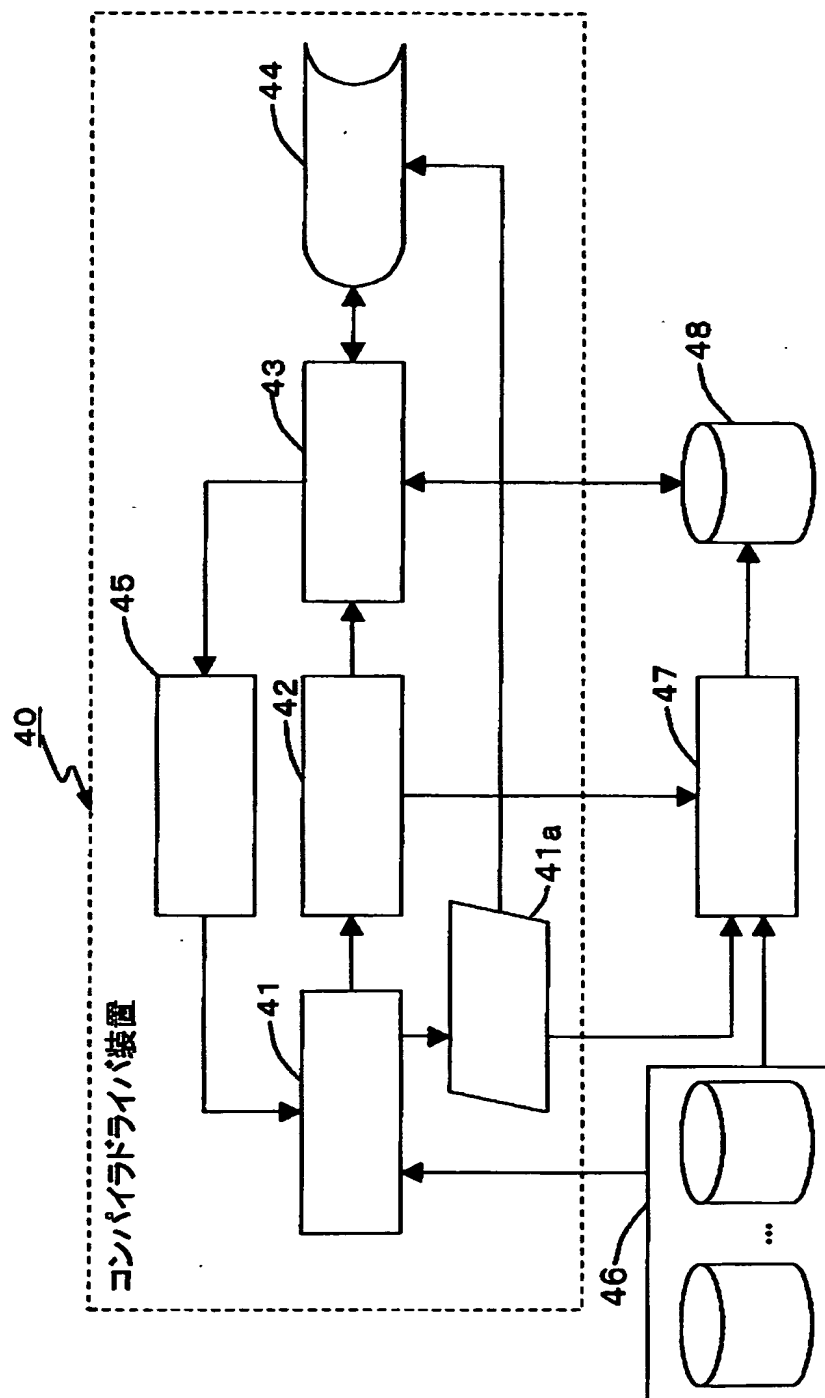
【図 3】



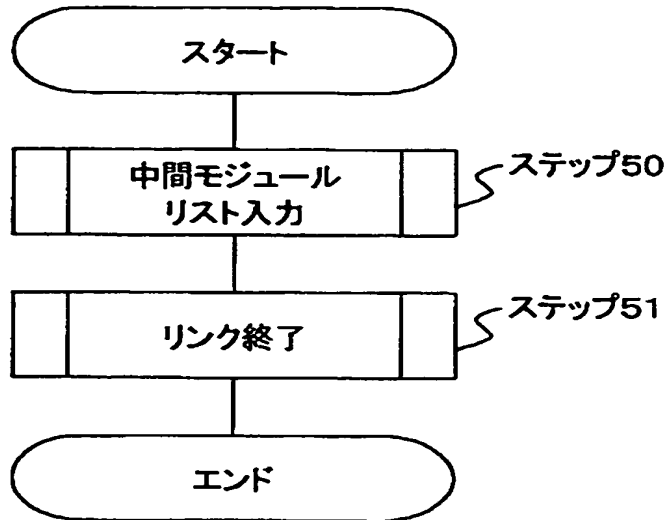
【図 4】



【図 5】



【図 6】



【書類名】 要約書

【要約】

【課題】 最小サイズの実行可能オブジェクトを容易に作成することのできる中間オブジェクト連結方法、及び、中間オブジェクト連結装置、並びに、連結するプログラムを記録した記憶媒体を提供する。

【解決手段】 連結順序作成部 1 1 は、遺伝的アルゴリズム等を適用して作成された連結順序に基づいて、リンカ起動部 1 2 がリンカ 1 7 を起動し、実行可能オブジェクト 1 8 を作成する。比較部 1 3 において、作成された実行可能オブジェクト 1 8 のプログラムサイズと、記憶部 1 4 に記憶している最小値とを比較し、プログラムサイズが最小値よりも小さい時は、記憶部 1 4 は、このプログラムサイズを最小値として記憶し、その時の連結順序も記憶する。繰り返し部 1 5 によってこの一連の工程が繰り返されることによって、人手を介さずに効率的に、プログラムサイズが最小となる連結順序が得られる。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [ 0 0 0 0 0 5 8 2 1 ]

1. 変更年月日	1 9 9 0 年 8 月 2 8 日
[変更理由]	新規登録
住 所	大阪府門真市大字門真 1 0 0 6 番地
氏 名	松下電器産業株式会社